

中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下：

This is to certify that annexed is a true copy from the records of this
office of the application as originally filed which is identified hereunder:

申請日：西元 2003 年 11 月 05 日
Application Date

申請案號：092130952
Application No.

申請人：財團法人工業技術研究院
Applicant(s)

局長
Director General

蔡練生

發文日期：西元 2004 年 1 月 6 日
Issue Date

發文字號：09320016300
Serial No.

發明專利說明書

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：

※ 申請日期：

※IPC 分類：

壹、發明名稱：(中文/英文)

自動服務組合方法及系統

METHOD AND SYSTEM OF AUTOMATIC SERVICE COMPOSITION

貳、申請人：(共 1 人)

姓名或名稱：(中文/英文)

財團法人工業技術研究院

INDUSTRIAL TECHNOLOGY RESEARCH INSTITUTE

代表人：(中文/英文) 翁政義 / WENG, CHENG-I

住居所或營業所地址：(中文/英文)

新竹縣竹東鎮中興路四段一九五號

No. 195, Sec. 4, Chung Hsing Rd., Chutung, Hsinchu

國 籍：(中文/英文) 中華民國 / R.O.C.

參、發明人：(共 3 人)

姓 名：(中文/英文)

1. 陳逸嘉 / CHEN, YI-CHIA

2. 許維德 / HSU, WEI-TEK

3. 洪鵬翔 / HUNG, PENG-HSIANG

住居所地址：(中文/英文)

1. 台北市松山區莊敬里6鄰撫遠街395巷3號4樓

4F., No.31, Lane 395, Fuyuan St., Songshan District, Taipei City

2. 台北縣土城市民族街20號4樓

4F., No.20, Minzu St., Tucheng City, Taipei County

3. 高雄市新興區開封路光耀里 22 鄰 90 號

No.90, Kaifong Rd., Sinsing District, Kaohsiung City

國 籍：(中文/英文) 1.2.3. 中華民國 / R.O.C.

肆、聲明事項：

☐ 本案係符合專利法第二十條第一項 ☐ 第一款但書或 ☐ 第二款但書規定之期間，其日期為： 年 月 日。

◎本案申請前已向下列國家（地區）申請專利 ☐ 主張國際優先權：

【格式請依：受理國家（地區）；申請日；申請案號數 順序註記】

1. 無

2.

3.

4.

5.

☐ 主張國內優先權（專利法第二十五條之一）：

【格式請依：申請日；申請案號數 順序註記】

1.

2.

☐ 主張專利法第二十六條微生物：

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

☐ 熟習該項技術者易於獲得，不須寄存。

伍、中文發明摘要：

本發明係有關於一種自動服務組合方法及系統，服務註冊者係將由服務提供者所傳來之服務規格儲存於服務資料庫中，並接收由服務需求者所提出之服務需求，俾以自服務資料庫中自動搜尋出適當的個別服務或流程組合以回傳給服務需求者據以執行。其中，服務註冊者係具有一組合引擎，其用以對上述服務規格及服務需求進行前處理，並宣告適當之新物件，俾便後續利用規劃引擎進行自動組合的運算；且對於規劃引擎所產生之解決方案，亦必須進行後處理，以轉換為流程語言，之後再交由流程引擎一一執行以完成服務需求。

陸、英文發明摘要：

A method and system for automatic service composition is disclosed, wherein a service registry saves a plurality of service specifications received from at least one service provider in a service repository, and receives service requests from service requestors. A composition engine of the service registry can automatically generate a series of actions from service specifications stored in the service repository according to the service request, and then passes those actions to the service requestor for being processed. In order to facilitate the proceeding procedure of automatic composition, the composition engine pre-processes service requests and service specifications, and declares suitable new objects. The composition engine also post-processes the planning resolution generated by the planning module for translating the format of the planning resolution into processing language, and finally passes the resolution to a processing engine of the service requestor for being processed.

柒、指定代表圖：

(一)本案指定代表圖為：圖(5)。

(二)本代表圖之元件代表符號簡單說明：

組合引擎 31	服務需求轉換模組 311
服務規格轉換模組 312	規劃模組 313
服務組合轉換模組 314	流程引擎 41
服務需求 R	服務規格 S
流程文件 P	問題敘述檔案 F1
領域敘述檔案 F2	服務組合流程檔案 F3

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

玖、發明說明：

【發明所屬之技術領域】

本發明係關於一種自動服務組合技術，尤指一種利用人工智慧(artificial intelligence, AI)之規劃技術(planning)的自動服務組合方法及系統，適用於網路環境中。

【先前技術】

按，隨著網際網路技術的不斷演進，促使網路服務(web services)越來越盛行，因此各種用以建置個別網路服務的相關通訊協定亦隨之應運而生，包括用於通訊的簡易物件存取協定(simple object access protocol, SOAP)、用於敘述的服務敘述語言工具(web services description language, WSDL)、以及用於探索和整合的普及敘述探索與整合工具(universal description discovery and integration, UDDI)，且上述通訊協定皆以可擴展標記語言(extensible markup language, XML)撰寫而成。

在可預見的未來，網路上的電子商務活動將會越趨複雜，一項交易可能牽涉到由複數個不同公司所提供的不同服務。例如欲經由網路服務來安排海外自助旅遊行程，則將可能牽涉到航空公司的機票服務、鐵路局的車票服務、及旅館的訂房服務...等，使用者必須由本地端電腦(local)逐一連線至提供該項網路服務的遠端網站來完成作業。此舉相當耗時費力，且容易因人為疏失而致使旅遊行程的完整性大打折扣，例如發生漏訂車票、或訂錯旅館日期的情

況等。因此，目前各界均致力於利用軟體來加強商業流程管理(business process management)，以促成網路自動化的使用環境。

一般網路自動化環境係包括自動服務組合和自動服務執行兩大功能。所謂自動服務組合就是使用者只要提出需求，系統就會去網路上自動地尋找合適的服務並加以組合成可以完成需求的流程；自動服務執行則是系統可根據流程所定義的步驟，自動地一一執行來滿足使用者需求。

請參閱圖 1，係為習知由產業界所提出用以在電子商務環境中執行網路服務組合的功能方塊圖。當程式設計者 P 接獲服務需求 R1 後，將透過流程設計工具 11(例如圖形使用者介面(graphic user interface, GUI)工具)以繪製出符合該服務需求 R1 的流程圖，並由流程設計工具 11 將此流程圖轉化為 XML 文件 X1，之後交由流程執行引擎 12(process execution engine)以根據 XML 文件 X1 的設定來一一執行對應動作，並且去呼叫不同的網路服務 S11, S12, S13。然而，由於上述方法僅藉由提供 GUI 工具或簡單的資料型態搜尋機制來協助程式設計者進行人工的服務，因此，程式設計者往往需要耗費非常多的時間去瞭解及尋找每一種不同網路服務的規則，之後才有辦法組合出符合各種服務需求的流程。此外，上述由程式設計者人工定義出的流程皆針對各種不同的需求所制訂出，故無法隨著外在環境的變動而自動調整，不具使用彈性及強健性(robustness)，並非十分理想。

圖 2 則為習知由學術界所提出之網路服務組合的方塊圖，其係由組合引擎 21(composition engine)根據服務需求 R2 自動產生流程所需的 XML 文件 X2，之後亦交由流程執行引擎 22 加以執行動作並呼叫網路服務 S21, S22, S23。

5 其中，組合引擎 21 係採用人工智慧(artificial intelligence, AI)的規劃(planning)技術所建構而成，並利用語意網路(semantic web)的 DAML-S 語言來加強對各種服務規格的敘述能力，例如定義服務的事前條件(precondition)及事後影響(effect)等。

10 然而，在上述系統所採用的 AI 規劃技術中，有所謂「新物件問題」：所有在流程中會使用到的相關物件皆必須於一開始就宣告出來，至於執行的動作(action)只是用以更改物件的狀態而已，並不能產生新物件(沒有在初始狀態宣告的物件)。但事實上，在執行網路服務的現實環境中產生新物件是常態，例如使用者發出需求想要得到 A 地到 B
15 地的旅遊行程，但使用者一開始並不知道安排旅程時可能會先查詢 A 地車站，產生車站物件 C，及查詢 B 地的車站，產生車站物件 D，所以一開始並未宣告車站物件 C 及車站物件 D，這會導致 AI 規劃技術無法正確產生服務流程的嚴重後果。由此可知，習知自動網路服務組合方法仍存在有
20 諸多缺失，無法應用於現實網路服務的環境，而有予以改進之必要。

【發明內容】

本發明之主要目的係在提供一種自動服務組合方法及系統，係藉由服務組合引擎(composition engine)自動且動態地尋找適當的服務、組合出一執行處理流程(process flow)，再交由流程引擎(process engine)一一執行以完成特定，俾以完成特定之服務需求。

本發明之另一目的係在提供一種自動服務組合方法及系統，係利用人工智慧(artificial intelligence, AI)之規劃(planning)技術、配合物件需求預估及宣告技術，以宣告在網路服務中可能產生的新物件，並對產生新物件之服務加上邏輯限制，以確保所宣告之新物件可正確地被使用，俾能加強自動服務組合技術的彈性與延伸性，並可減少複雜的系統整合時間及金錢。

依據本發明之一特色，所提出之自動服務組合方法，用以根據一服務需求(service request)、以自至少一已註冊之服務規格(service specification)中自動搜尋並組合出符合該服務需求之個別服務或服務流程組合。本發明首先係接收由一服務需求所建立之問題敘述檔案(problem file)，及由複數個服務規格所建立之領域敘述檔案(domain file)。在問題敘述檔案或領域敘述檔案內，皆可宣告物件。服務需求係利用這些物件，以邏輯運算式(logic statements)定義一初始狀態(initial state)及一終止狀態(goal state)，俾供取得一連串動作(action)的執行順序來將相關物件從初始狀態轉換至終止狀態而達成服務需求的定義；每一服務規格係用以描述每一行動，其定義有一行動名稱、零或多個輸入參數、零或多個輸出參數，每一參數皆定義有一參

數名稱及一資料型態(data type)，並利用這些參數以邏輯運算式定義零或多個執行前條件(precondition)、及零或多個執行後狀況(effect)。在利用規劃技術找尋可執行流程時，會反覆的判斷某一行動是否需要擺在結果流程內及要擺在結果流程的那個時間點。在做此判斷時，該行動的每一參數都必須連結(bind)到與其相同資料型態的物件。如果有一物件集合能夠連結到該行動的每一個參數，且該物件集合的狀態符合該行動的執行前條件，規劃引擎就會依照此行動的執行後狀況更新該物件集合內物件的狀態。因此如果沒有宣告一個在流程內必要的行動所需的物件，這可能會導致最後無法產生完整的行動流程，這也是導致AI規劃技術的「新物件問題」的主要原因。

為了解決習知AI規劃技術的「新物件問題」，本發明藉由分析服務規格及服務需求，預測並宣告最後完整流程中的網路服務所可能產生及使用的新物件。首先，根據服務規格內之定義，建立至少一層所有服務規格與資料型態間之交互使用關連性結構；之後即擷取出服務需求所宣告之物件需使用到的資料型態，以選取出與資料型態相關的服務規格，並將選出的服務規格皆儲存在一選定服務集合中；再來，則需統計在選定服務集合中、每一種資料型態作為輸入參數的次數 N ，每一種資料型態作為輸出參數的次數 M ；若 $M > 0$ 即此資料型態亦需作為輸出參數、則針對此資料型態與領域敘述檔案中對應宣告 $C \times (N + M)$ 個新物件；最後，將根據問題敘述檔案及領域敘述檔案的定義以產生一具有一連串動作之執行順序的服務組合流程，俾供

加以執行而達成服務需求。本發明並可遵循前述交互使用
關連性結構，藉以選取出與資料型態相關的服務規格，以
將選出的服務規格皆儲存在一選定服務集合。

依據本發明之另一特色，所提出之自動服務組合系統
5 主要包括一服務資料庫、一組合引擎、及一流程引擎，且
其中之組合引擎係由一服務需求轉換模組、一服務規格轉
換模組、一規劃模組、及一服務組合轉換模組所組成。本
發明之自動服務組合系統係用以執行上述自動服務組合方
法所主張之步驟，以根據一服務需求、自複數個已註冊之
10 服務規格中自動搜尋並組合出符合上述網路需求的個別服
務或服務流程組合。

【實施方式】

為能讓 貴審查委員能更瞭解本發明之技術內容，特
15 舉較佳具體實施例說明如下。

請先參閱圖3本發明之實施環境示意圖，其係運用網
路服務及語意網路之相關技術來建構出一分散式運算環
境，俾利進行實作。如圖3所示，本實施例係由服務註冊者
(service registry)3、服務需求者(service requester)4、及服
20 務提供者(service provider)5建立一套延伸網路服務架構。

服務提供者5係將其欲在網路環境中所提供的各種網
路服務明確定義出各個服務規格(service specification)，例
如航空業者所提供之網路訂機票服務、鐵路局所提供之網
路訂車票服務、旅館業者所提供之網路訂房服務、及/或氣
25 象局所提供之網路天氣查詢服務...等，之後透過普及敘述

探索與整合(universal description discovery and integration, UDDI)通訊協定以將上述定義好的服務規格公布(publish)到服務註冊者3來完成註冊。當服務註冊者3接收到由各個不同的服務提供者5所傳來之服務規格時，即將這些服務規格的相關資訊儲存於服務資料庫(service repository)32中；此外，服務註冊者3亦可接收由服務需求者4所提出的服務需求(service request)，以由組合引擎31自服務資料庫32中自動且動態地搜尋出適當的個別服務、或由多個服務所組成的流程組合以回傳給服務需求者4，交由流程引擎41加以執行來完成服務需求。需注意的是，組合引擎31及流程引擎41亦可依系統需求而獨立成為一模組，並不限於本實施例之實施態樣。而服務需求者4亦可以是程式發展者(developer)或一般使用者(user)。

其中，本實施例由服務提供者5所定義之服務規格係用以執行一行動(action)，其包括有：服務位址、行動名稱、輸入參數、輸出參數、執行前條件(precondition)、及執行後狀況(effect)；而由服務需求者4所提出之服務需求則包括有一初始狀態(initial state)、及一終止狀態(goal state)，俾供取得一連串動作的執行順序來將初始狀態轉換至終止狀態而達成服務需求。服務需求將被轉換成問題敘述檔案，複數個服務規格將被轉換成領域敘述檔案。描述服務需求及服務規格所需的物件可以宣告於問題敘述檔案或領域敘述檔案內，其定義包括有一物件名稱及一資料型態(data type)。

服務需求內的初始狀態和終止狀態，及服務規格內的執行前條件和執行後狀況，都是用邏輯敘述式(logic statements)加以描述。一個邏輯敘述式是由一個或數個基本計算式(atomic formula)及零個或數個邏輯運算子(logic operator)所組成。邏輯運算子就是and、or和not。基本運算式則是由述詞(predicate)和一個或數個物件(或參數)名稱所組成，用來表達物件(參數)自己本身或和其他物件(參數)具有述詞所定義的屬性，例如：在Start[locObj1]中，Start就是一個述詞，locObj1是一個物件，Start[locObj1]表示地點locObj1具有Start這個屬性，可解釋為地點locObj1是一個起使地點；又，下列基本運算式：

Trip [itineraryObj locObj1 locObj2]

則表示itineraryObj，locObj1，locObj2和三個物件具有Trip這個屬性，可解釋為在itineraryObj所儲存的資料是從locObj1到locObj2的行程安排；而：

Start[locObj1] and End[locObj2]

就是包含一個邏輯運算子及兩個基本運算式的邏輯運算式。邏輯運算式在不同的計算機語言可能有不一樣的描述方式，但意義是相近的。例如在PDDL內，上述的基本運算式是表達成：

(and (Start locObj1) (End locObj2))

有些邏輯運算式可能省略邏輯運算子，而用and或是or當成預設的邏輯運算子連接在同一個邏輯運算式內的所有基本運算式，例如：PDDL問題敘述檔案內的初始條件就是以and當成預設的邏輯運算子。

請參閱圖4關於服務規格之示意圖，於本實施例中，服務資料庫32係儲存有m個由各個不同服務提供者5所定義之服務規格集合S，以第i個服務規格 S_i ($1 \leq i \leq m$)為例，其需使用到p個輸入參數、及n個輸出參數，且每一輸入參數與輸出參數皆對應定義有一資料型態，例如服務規格 S_i 的第k個輸入參數 $iObj_{(i,k)}$ ($1 \leq k \leq p$)所對應之資料型態為 $iType_{(i,k)}$ 、服務規格 S_i 的第j個輸出參數 $oObj_{(i,j)}$ ($1 \leq j \leq n$)所對應之資料型態為 $oType_{(i,j)}$ 。上述輸入參數、輸出參數、資料型態、執行前條件、及執行後狀況之詳細定義將於後文中一一詳述。又，本實施例係採用DMAL-S及WSDL語言標準來定義每一組服務規格，其中，DAML-S係用以提供較抽象的服務規格，包括執行前條件、及執行後狀況之定義敘述；WSDL則用以提供較具體化的服務規格，包括服務位址、行動名稱、輸入參數、及輸出參數等定義敘述。由於DAML-S及WSDL係採用資源描述架構(resource description framework, RDF)和可擴展標記語言(extensible markup language, XML)的語法，故每個標籤皆具有明確的定義，俾利於後續轉換過程。需注意的是，描述服務規格的語言並不限於本實施例所使用的DAML-S及WSDL。只要能提供相關欄位的語言或規格，皆可應用於本發明。

接下來請一併參閱圖5之功能方塊圖及圖6之流程圖。本實施例之組合引擎31首先係接收儲存在服務資料庫32內的各種服務規格S(步驟S601)，之後交由服務規格轉換模組(service translator)312以將原本由XML格式撰寫而成的服務規格S轉換為符合規劃領域定義語言(planning

domain definition language, PDDL)格式之領域敘述檔案 F2(domain file)(步驟S602)，用以在PDDL中描述各個行動的規格。而當服務註冊者3接收到由服務需求者4所傳送過來之服務需求R時(步驟S603)，亦需透過服務需求轉換模組 311以將XML格式之服務需求R轉換為PDDL格式之問題敘述檔案F1(步驟S604)。接著，上述之問題敘述檔案F1及領域敘述檔案F2將傳送給規劃模組313，藉以運算出可滿足服務需求R的一連串服務規格S所形成的服務組合流程檔案F3(步驟S605)，之後同樣並透過服務組合轉換模組314以將PDDL格式的服務組合流程檔案F3轉換為XML格式之流程文件P(步驟S606)，最後傳送回服務需求者4端的流程引擎41據以執行(步驟S607)。需注意的是，本發明並不限於將服務需求和服務規格轉換成PDDL，只要能提供相等訊息的語言，皆可為轉換的標的語言。

如上所述，由於RDF(S)和XML語法對於每個標籤都具有明確的定義，且能夠輕易對應到PDDL之基本語言單元，因此本實施例之服務需求轉換模組311、服務規格轉換模組312、及服務組合轉換模組314在進行將XML檔案轉換為PDDL檔案、或將PDDL檔案轉換為XML檔案之流程均為熟習該項技術者所熟知，故不在此贅述該轉換方式。

接著請參閱圖7，其係為圖6之步驟S602中、用以將服務規格S轉換為領域敘述檔案F之細部流程，以下並配合本實施例之服務需求R及服務規格S加以說明，俾以解決在規劃服務組合的過程中、產生新物件的問題。於本實施例中，服務需求者4所提出之服務需求R的目的在於「尋找兩個地

點間的旅遊安排」，例如為欲自台北到高雄的旅遊行程，因此，服務需求R係於初始狀態和終止狀態分別預先宣告有下列物件，及利用述詞定義物件的屬性：

(1)初始狀態：Location locObj1；

5 Location locObj2；
 Itinerary itineraryObj;
 Start [locObj1]；
 End [locObj2]

(2)終止狀態：Trip [itineraryObj locObj1 locObj2]

10 其中，上述物件所涵蓋的資料型態包含有「Location」及「Itinerary」兩種，分別代表「地點」及「旅行計畫」。而在初始狀態所宣告兩個資料型態分別為「Location」的物件中，利用述詞「Start」定義「Start[locObj1]」這個基本運算式，表示物件「locObj1」為起始地、利用述詞「End」
15 定義「End[locObj2]」這個基本運算式，表示物件「locObj2」為終止地。故依據本例之服務需求R，可得知物件「locObj1」即代表起始地台北、而物件「locObj2」則代表終止地高雄；終止狀態則利用述詞「Trip」定義基本運算式如下：

Trip [itineraryObj locObj1 locObj2]，

20 其係表示物件「itineraryObj」所存的資料是「locObj1」到「locObj2」的旅行計畫。

由於服務需求者4無從得知是否會在執行上述服務需求R的過程中產生新物件，原則上也只有在產生了完整的服務流程後，才可能確定地知道這整個流程中會使用到哪
25 些新物件，但如果一開始沒有宣告新物件，也無法產生完

整流程。因此本發明根據圖7之流程分析服務需求R及服務規格S間之關連性，俾預測並宣告可能產生的新物件。為了節省組合引擎31的運算能力和系統資源，盡量只宣告和完成與服務需求R相關的新物件，首先必須預測服務需求R可能會使用到哪些服務，才能再預測出這些服務可能會產生的新物件的資料型態與個數。因此根據圖7之流程，分析領域敘述檔案F2之定義，以建立k層由所有服務規格S與資料型態間之交互使用關連性結構圖(data-type-service graph)(步驟S701)，其較佳可於離線狀態下進行、以減少網路負荷量，當然亦可於連線狀態下進行。如圖8所示，本實施例係建立二層(k=2)之結構圖，其亦可根據系統性能而調整欲建立之層級結構數。此外，為簡化結構，故本實施例之服務規格集合S僅包含「FindStation」及「MakeItinerary」二種(即m=2)，其服務規格分別定義為：

FindStation (Location ?loc, Station ?station) :

執行前條件：無定義；

執行後狀況：At [?station,?loc]；

MakeItinerary (Location ?loc1, Location ?loc2, Station ?station1, Station ?station2, Itinerary ?i)

執行前條件：Start[?loc1] and End [?loc2] and At [?station1, ?loc1] and At [?station2, ?loc2]

執行後狀況：Trip [?i, ?loc1, ?loc2]

其中，以”?”開頭的名稱代表服務規格所定義的輸入或輸出參數。服務規格「FindStation」係用以在輸入地點

「Location ?loc」(即輸入參數)後、找出鄰近機場(或車站)

「 Station ?station 」 (即 輸 出 參 數) ； 而 服 務 規 格
「 MakeItinerary 」 則 用 以 在 輸 入 兩 個 地 點 (「 Location ?loc1 」
和 「 Location ?loc2 」) ， 及 兩 個 車 站 (「 Station ?station1 」
和 「 Station ?station2 」) ， 而 且 若 符 合 執 行 前 條 件 ：

5 Start[?loc1] (loc1 是 起 始 地 點) 、
 and At [?stataion1, ?loc1] (且 ?station1 是 ?loc1 的 的 車
 站) 、

 and End[?loc2] (且 ?loc2 是 終 止 地 點) 、

 and At[?station2, ?loc2] (且 ?station2 是 ?loc2 的 車 站) ，

10 如 此 一 來 ， 根 據 執 行 後 狀 況 Trip [?i, ?loc1, ?loc2] ， 此 服 務
 就 會 提 供 一 套 從 地 點 「 ?loc1 」 到 地 點 「 ?loc2 」 的 行 程 「 ?i 」 。
 當 然 於 實 際 運 用 中 將 會 提 出 各 式 各 樣 的 服 務 規 格 集 合 S ，
 並 不 僅 限 於 上 述 範 疇 。

 於 圖 8 中 ， 係 先 擷 取 出 儲 存 在 服 務 資 料 庫 32 中 所 有 服
15 務 規 格 所 牽 涉 到 的 資 料 型 態 ， 即 「 Location 」 、 「 Station 」
 及 「 Itinerary 」 ， 再 逐 一 找 出 與 每 個 資 料 型 態 相 關 的 服 務
 規 格 ， 例 如 在 第 一 層 結 構 中 ， 資 料 型 態 「 Inineraray 」 只 被
 服 務 規 格 「 MakeItinerary 」 使 用 到 ， 因 此 將 在 兩 者 間 建 立
 起 一 道 關 連 性 ， 同 理 ， 資 料 型 態 「 Location 」 和 「 Station 」
20 亦 與 其 相 關 的 服 務 規 格 間 建 立 有 關 連 性 ； 接 著 第 二 層 結 構
 則 將 各 服 務 規 格 所 會 使 用 到 的 物 件 關 係 建 立 起 來 ， 例 如 服
 務 規 格 「 MakeItinerary 」 牽 涉 三 種 資 料 型 態 「 Itinerary 」 、
 「 Location 」 及 「 Station 」 ， 餘 此 類 推 。

 接 著 ， 將 根 據 服 務 需 求 R 所 宣 告 到 的 資 料 型 態 ， 自 上
25 述 結 構 圖 中 找 出 與 被 宣 告 之 資 料 型 態 相 關 的 服 務 規 格 (步

驟 S702)，並將選出的服務規格儲存在一選定服務集合 (choosed service set) 中。例如在服務需求 R 中宣告有資料型態「Itinerary」，由結構圖第一層可看出「Itinerary」被服務規格「MakeItinerary」使用到，但當延伸到結構圖第二層時，發現「MakeItinerary」一共使用到「Itinerary」、「Location」和「Station」三種資料型態，且「Station」與服務規格「FindStation」相關。故由此可知，若 $k=1$ ，則與資料型態「Itinerary」相關的服務規格只有「MakeItinerary」，但在 $k=2$ 的設定下與資料型態「Itinerary」相關的服務規格除了「MakeItinerary」之外，還包括有「FindStation」。因此 k 如果越大，則可以推算出越多和服務需求相關的服務，選定服務集合內所包含的服務就越多，自動服務組合成功的機率也越高，但也佔用組合引擎 31 越多的運算及系統資源。所以 k 可以根據組合引擎 31 的系統狀況做動態調整。

之後，將統計出現在選定服務集合中，每一種資料型態作為輸入參數的出現總次數 $N(t)$ 、及輸出參數的出現總次數 $M(t)$ (步驟 S703)。由於在本實施例之假設中，只要有輸出參數就代表可能產生新物件，因此當某一資料型態作為輸出參數之次數 $M(t) > 0$ 時 (步驟 S704)，代表該資料型態亦被作為輸出參數，故將在 PDDL 領域敘述檔案 F2 中宣告 $C \times (N(t) + M(t))$ 個新物件 (步驟 S705)，其中 C 為一整數，其代表組合引擎 31 在一次規劃的過程內，最多可以接受一個行動被呼叫幾次，即由系統管理者根據組合引擎 31 所在的機器運算能力或系統負荷，來動態調整 C 。據此，將可計算

出本實施例之選定服務集合中、各個資料型態作為輸入參數及輸出參數的次數如下：

Location : $N(t) = 3$ 、 $M(t) = 0$ ；

Station : $N(t) = 2$ 、 $M(t) = 1$ ；

5 Itinerary : $N(t) = 0$ 、 $M(t) = 1$ 。

於上述資料型態中，以「Station」為例，它作為輸出參數的次數大於0，若假設 $C = 1$ ，則本實施例將宣告 $3(C \times (N(t) + M(t))) = 1 \times (2 + 1) = 3$ 個資料型態為「Station」的新物件，分別為「Station newObj1」、「Station newObj2」及
10 「Station newObj3」。

又，為了防止未來新宣告的物件會被錯誤的重複利用，如圖9，執行兩次FindStation，雖然輸入不同的地點locOb1和locOb2，但若不加以限制，最後都將找出的車站存入newObj1，雖然仍可能產生一個可執行的流程，但是因為變數被重複指定，最後的執行結果就可能會出錯。因此
15 必須針對每一輸出參數 $oObj_{(i,j)}$ 多加一組執行前條件(not (used $oObj_{(i,j)}$))、和執行後狀況(used $oObj_{(i,j)}$)的邏輯敘述(步驟S706)。多加的執行前條件可以限制只有沒有被用過的新物件才可以被利用；而多加的執行後狀況則可以明確
20 宣告哪一個新物件已經被利用過了，才不會被未來其他要產生同樣資料型態的新物件之行動所誤用。由於各服務規格S皆定義有執行前條件與執行後狀況，因此每個服務規格係環環相扣，且每個行動的參數皆必須連結到一個已經存在的物件才能執行後續規劃。如此一來，規劃模組313
25 即可根據由服務需求R所組成的問題敘述檔案F1、以及由

服務規格S併同服務需求R所組成的領域敘述檔案F2，來規劃出由一連串動作所組成的服務組合流程檔案F3(即圖6之步驟S605)，其包括：

```
FindStation(Location ?loc, Station ?station) :  
5      Location ?loc = locObj1、  
      Station ?station = newObj1 ;  
FindStation(Location ?loc, Station ?station) :  
      Location ?loc = locObj2、  
      Station ?station = newObj2 ;  
10  MakeItinerary (Location ?loc1, Location ?loc2,  
      Station ?station1, Station ?station2, Itinerary ?i)  
      Location ?loc1=locObj1、  
      Location ?loc2=locObj2、  
      Station ?station1= newObj1、  
15      Station ?station2= newObj2、  
      Itinerary ?i = itineraryObj。
```

亦即流程引擎41必須先執行兩次「FindStation」以分別找出鄰近台北的車站(機場)「newObj1」、和鄰近高雄的車站(機場)「newObj2」，之後再執行「MakeItinerary」，
20 用以由台北到高雄之行程。

圖10係為由服務需求轉換成的PDDL問題敘述檔案之程式碼，圖11則是沒有利用圖7之流程而將服務規格直接轉換出的PDDL領域敘述檔案之程式碼。如果將圖10和圖11的PDDL檔案輸入給AI規劃引擎，則將會因為沒有宣告新
25 物件的問題，而無法產生完整的服務流程。而圖12即是利

用本發明圖7之流程分析服務規格及服務需求所轉換出的PDDL領域敘述檔案之程式碼，因為其中有事先自動宣告可能會產生的新物件，所以可以產生完整服務流程，進而完成自動服務組合。

- 5 根據上述之說明，顯示本發明所提出之自動網路服務組合方法係藉由人工智慧的規劃技術，以根據使用者或發展者的網路服務需求，透過服務組合引擎來自動且動態地尋找適當的網路服務、並組合出一個可以完成任務的執行流程，再交給流程引擎一一執行。本發明並配合物件預估
- 10 及宣告技術，以宣告每個服務可能產生的新物件，俾能解決習知無法產生新物件的問題，確保每一筆服務需求皆可獲得完善的網路服務，實為一大進步。

- 上述實施例僅係為了方便說明而舉例而已，本發明所主張之權利範圍自應以申請專利範圍所述為準，而非僅限
- 15 於上述實施例。

【圖式簡單說明】

- 圖1係習知產業界所提出之網路服務組合之功能方塊圖。
- 圖2係習知學術界所提出之網路服務組合之功能方塊圖。
- 圖3係本發明較佳實施例之實施環境示意圖。
- 20 圖4係本發明較佳實施例之服務規格之示意圖。
- 圖5係本發明較佳實施例之功能方塊圖。
- 圖6係本發明較佳實施例之流程圖。
- 圖7係本發明較佳實施例將服務規格轉換為領域敘述檔案之流程圖。

圖8係本發明較佳實施例之交互使用關連性結構之示意圖。

圖9係本發明較佳實施例中、新宣告物件被重複利用之交互使用關連性結構之示意圖。

圖10係本發明較佳實施例由服務需求所轉換成之PDDL問題敘述檔案之程式碼。

圖11係本發明較佳實施例未利用圖7之流程而將服務規格直接轉換出之PDDL領域敘述檔案之程式碼。

圖12係本發明較佳實施例利用圖7之流程分析服務規格及服務需求所轉換出之PDDL領域敘述檔案之程式碼。

10

【圖號說明】

程式設計者 D

流程設計工具 11

流程執行引擎 12

組合引擎 21

流程執行引擎 22

服務需求 R1, R2

15 XML 文件 X1, X2

網路服務 S11, S12, S13, S21, S22, S23

服務註冊者 3

組合引擎 31

服務需求轉換模組 311

服務規格轉換模組 312

規劃模組 313

服務組合轉換模組 314

20 服務資料庫 32

服務需求者 4

流程引擎 41

服務提供者 5

服務需求 R

服務規格 S

流程文件 P

問題敘述檔案 F1

領域敘述檔案 F2

服務組合流程檔案 F3



拾、申請專利範圍：

1. 一種自動服務組合方法，自至少一已註冊之服務規格中自動搜尋並組合出符合一服務需求之個別服務或服務流程組合，該方法包括：

5 一接收服務需求步驟，係接收由該服務需求所建立之問題敘述檔案；

 一接收服務規格步驟，係接收由至少一服務規格所建立之領域敘述檔案，該至少一服務規格係用以執行一行動，其定義有一行動名稱、零或至少一輸入參數、零或至少一輸出參數，其中，任二相異之服務規格係可使用具有
10 相同資料型態之物件作為輸入參數或輸出參數；

 一新物件預估步驟，係擷取出被該問題敘述檔案或該領域敘述檔案宣告之物件需使用之資料型態，以選取出與該資料型態相關之至少一服務規格，並將該等選出之服務
15 規格儲存於一選定服務集合中；

 一新物件宣告步驟，係統計該選定服務集合中、每一種資料型態作為輸入參數的次數 N ，每一種資料型態作為輸出參數的次數 M ，若 $M > 0$ 即此資料型態亦需作為輸出參數，則針對該資料型態於該領域敘述檔案中宣告 $C \times (N + M)$
20 個新物件，其中， C 為一整數；以及

 一產生服務組合步驟，係根據該問題敘述檔案及該領域敘述檔案以產生一具有一連串動作之執行順序的個別服務或服務組合流程，俾供加以執行而達成該服務需求。

2. 如申請專利範圍第1項所述之方法，其中包括一建立關連性步驟，係根據該服務規格內之定義，以建立至少一層所有服務規格與資料型態間之交互使用關連性結構，而其中該物件預估步驟，係遵循該交互使用關連性結構，以選取出與該資料型態相關之至少一服務規格。

3. 如申請專利範圍第1項所述之方法，其中該服務需求係定義有一初始狀態及一終止狀態，而其中係以前述一連串動作的執行順序來將該初始狀態轉換至該終止狀態而達成該服務需求。

4. 如申請專利範圍第1項所述方法，其中，該服務需求係為可擴展標記語言(XML)格式之檔案，並經由一轉換程序以轉換為規劃領域定義語言(PDDL)格式之問題敘述檔案。

5. 如申請專利範圍第1項所述之方法，其中，該等服務規格係為XML格式之檔案，並經由一轉換程序以轉換為PDDL格式之領域敘述檔案。

6. 如申請專利範圍第5項所述之方法，其係使用DMAL-S規格以將該等服務規格定義為以RDF格式為基礎之檔案。

7. 如申請專利範圍第5項所述之方法，其係使用WSDL規格以將該等服務規格定義為XML格式之檔案。

8. 如申請專利範圍第1項所述之方法，其中，該等服務規格係由至少一服務提供者所定義，並公布至一服務註冊者之服務資料庫中。

9. 如申請專利範圍第8項所述之方法，其係使用 UDDI 註冊協定以將該等服務規格公布至該服務註冊者。

10. 如申請專利範圍第1項所述之方法，其中，該服務規格定義有零或至少一執行前條件、及零或至少一執行後狀況。

11. 如申請專利範圍第10項所述之方法，其中，於該物件宣告步驟後，更包括一修正步驟，用以對該選定服務集合中、每一服務規格之每一輸出參數新增一執行前條件、及一執行後狀況。

10 12. 一種自動服務組合系統，自至少一已註冊之服務規格中自動搜尋並組合出符合一服務需求之個別服務或服務流程組合，該自動服務組合系統係包括：

一服務資料庫，係由一服務註冊者所管理，並儲存有至少一服務規格，每一服務規格係用以執行一行動，其定義有一行動名稱、零或多個輸入參數、零或多個輸出參數，
15 其中，任二相異之服務規格係可使用具有相同資料型態之物件作為輸入參數或輸出參數；

一組合引擎，其包括有：

一服務需求轉換模組，係將前述服務需求轉換為
20 一問題敘述檔案；

一服務規格轉換模組，係將服務規格轉換為一領域敘述檔案，擷取出該服務需求所宣告之物件需使用之資料型態，並選取出與該資料型態相關之至少一服務規格，再將該等選出之服務規格儲存於一選定服務
25 集合中，之後統計於該選定服務集合中，每一種資料

型態作為輸入參數的次數N，每一種資料型態作為輸出參數的次數M，若 $M > 0$ 即此資料型態亦需作為輸出參數、則針對該資料型態於該領域敘述檔案中宣告 $C \times (N+M)$ 個新物件，其中，C為一整數；

5 一規劃模組，係根據該問題敘述檔案及該領域敘述檔案以產生一具有一連串動作之執行順序的個別服務或服務組合流程，俾供加以執行而達成該服務需求；及

10 一服務組合轉換模組，係將該服務組合流程轉換為一流程文件；

以及

一流程引擎，係執行該流程文件所定義之服務規格以達成該服務需求。

15 13. 如申請專利範圍第12項所述之系統，其中，該組合引擎係儲存於一服務註冊者中。

14. 如申請專利範圍第12項所述之系統，其中，該等服務規格係由至少一服務提供者所定義，並公布至該服務註冊者之服務資料庫中。

20 15. 如申請專利範圍第12項所述之系統，其中，該服務需求轉換模組係將XML格式之服務需求轉換為PDDL格式之問題敘述檔案。

16. 如申請專利範圍第12項所述之系統，其中，該服務規格轉換模組係將XML格式之服務規格轉換為PDDL格式之領域敘述檔案。

17. 如申請專利範圍第16項所述之系統，其係使用DMAL-S規格以將該等服務規格定義為以RDF格式為基礎之檔案。

5 18. 如申請專利範圍第16項所述之系統，其係使用WSDL規格以將該等服務規格定義為XML格式之檔案。

19. 如申請專利範圍第12項所述之系統，其中，該服務組合轉換模組係將PDDL格式之服務組合流程轉換為XML格式之流程文件。

10 20. 如申請專利範圍第12項所述之系統，其中，該服務規格定義有零或至少一執行前條件、零或至少一執行後狀況。

21. 如申請專利範圍第20項所述之系統，其中，該服務規格轉換模組尚包括用以對該選定服務集合中、每一服務規格之每一輸出參數新增一執行前條件、及一執行後狀況。

15 22. 如申請專利範圍第12項所述之系統，其中，該服務規格轉換模組，係根據該服務規格內之定義，以建立至少一層所有服務規格與資料型態間之交互使用關連性結構，並遵循該交互使用關連性結構，以選取出與該資料型態相關之至少一服務規格。

20 23. 如申請專利範圍第12項所述之系統，其中該服務需求係定義有一初始狀態及一終止狀態，而其中係以前述一連串動作的執行順序來將該初始狀態轉換至該終止狀態而達成該服務需求。

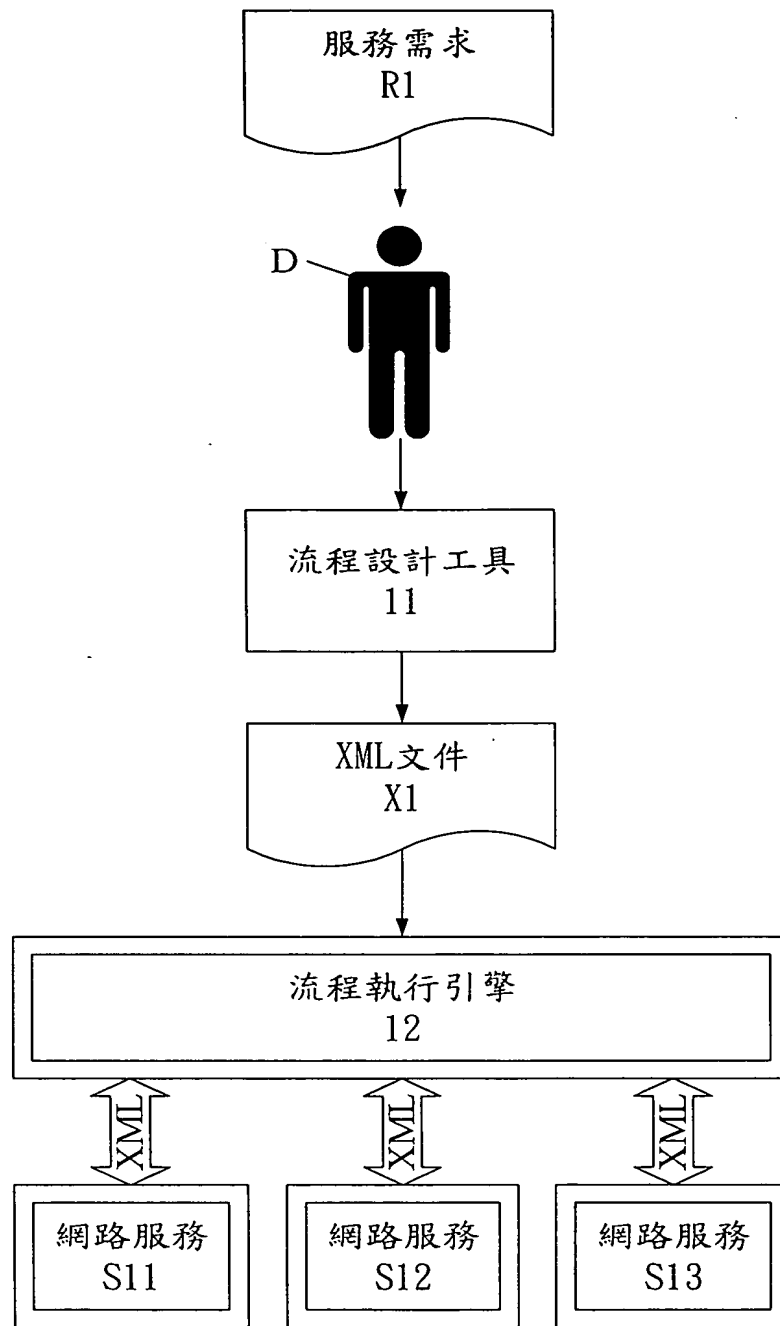


圖 1

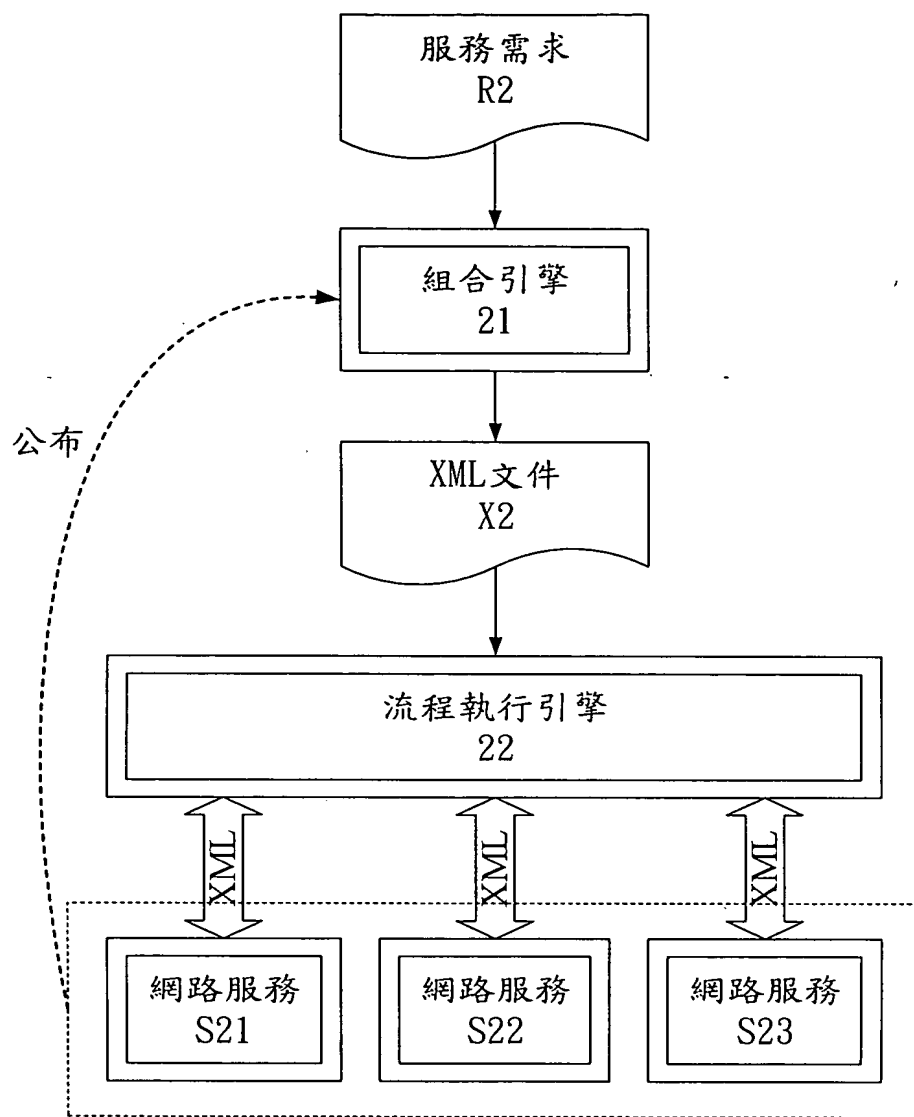


圖 2

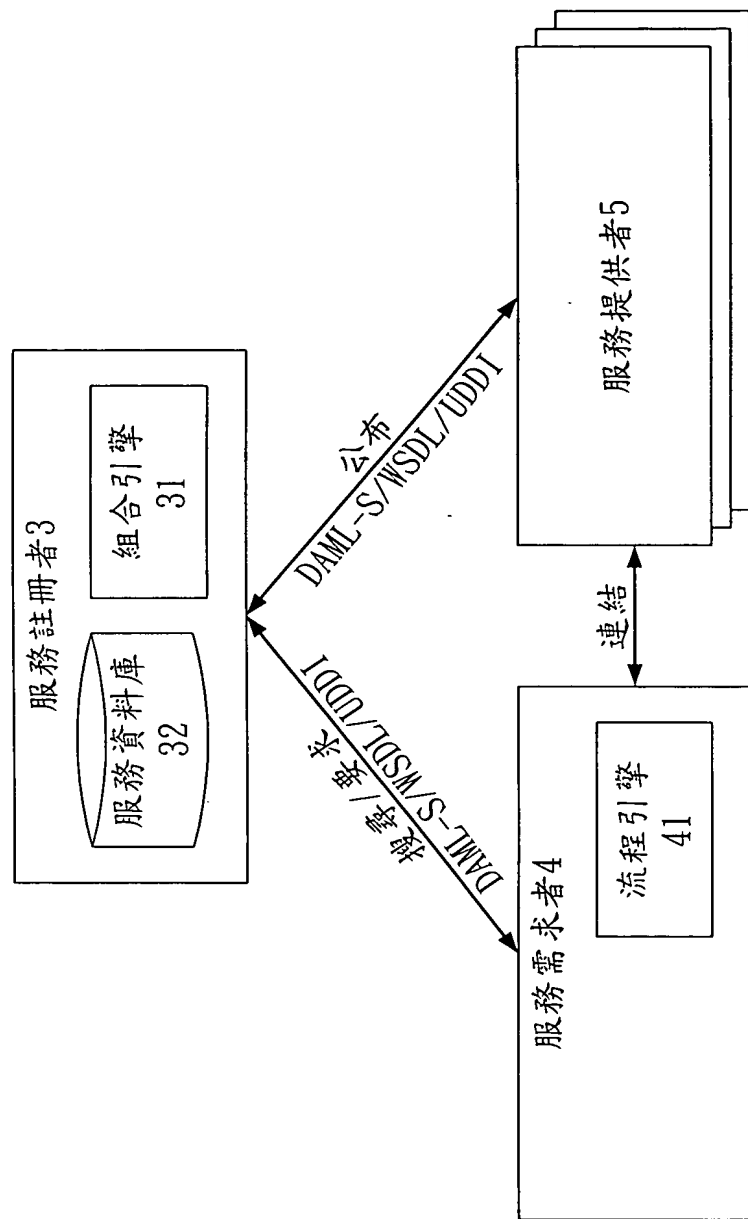


圖3

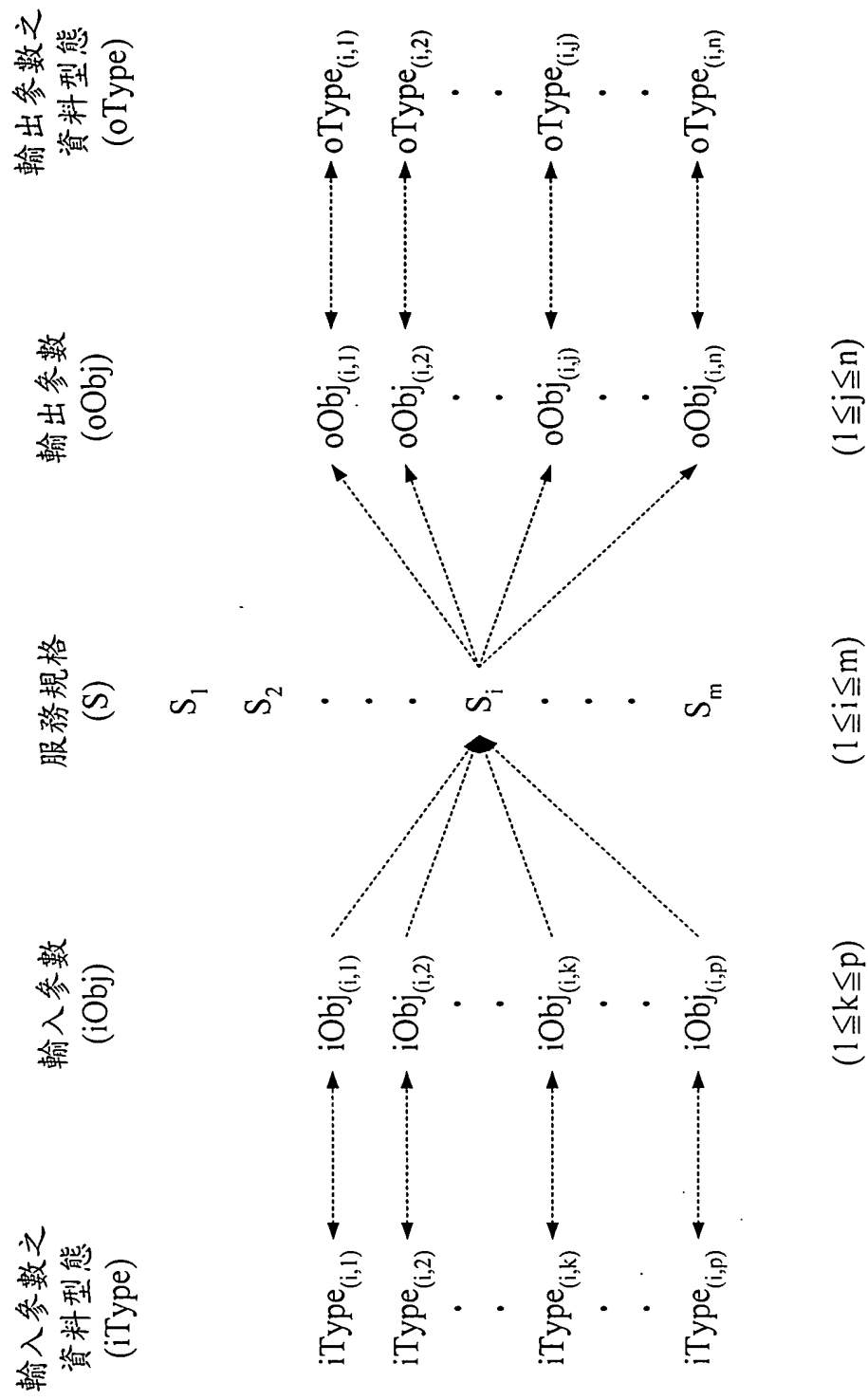


圖 4

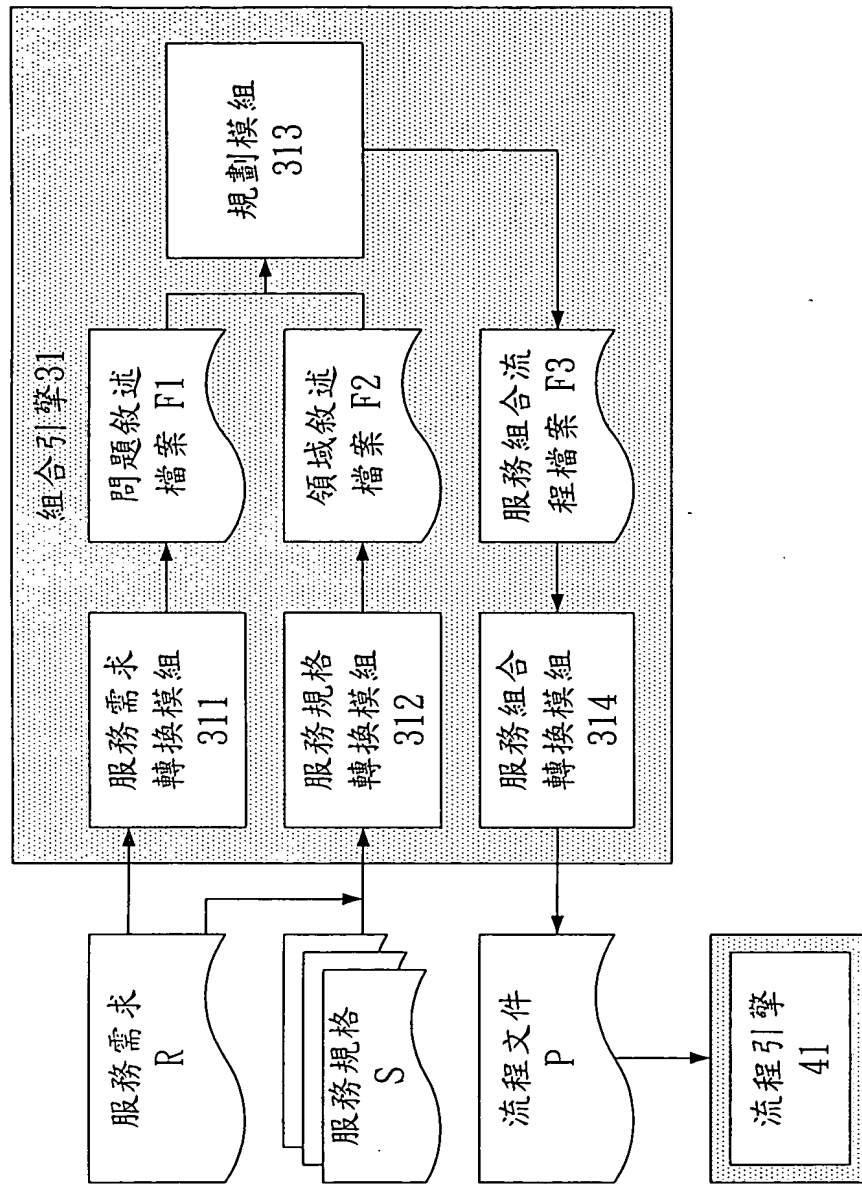


圖5

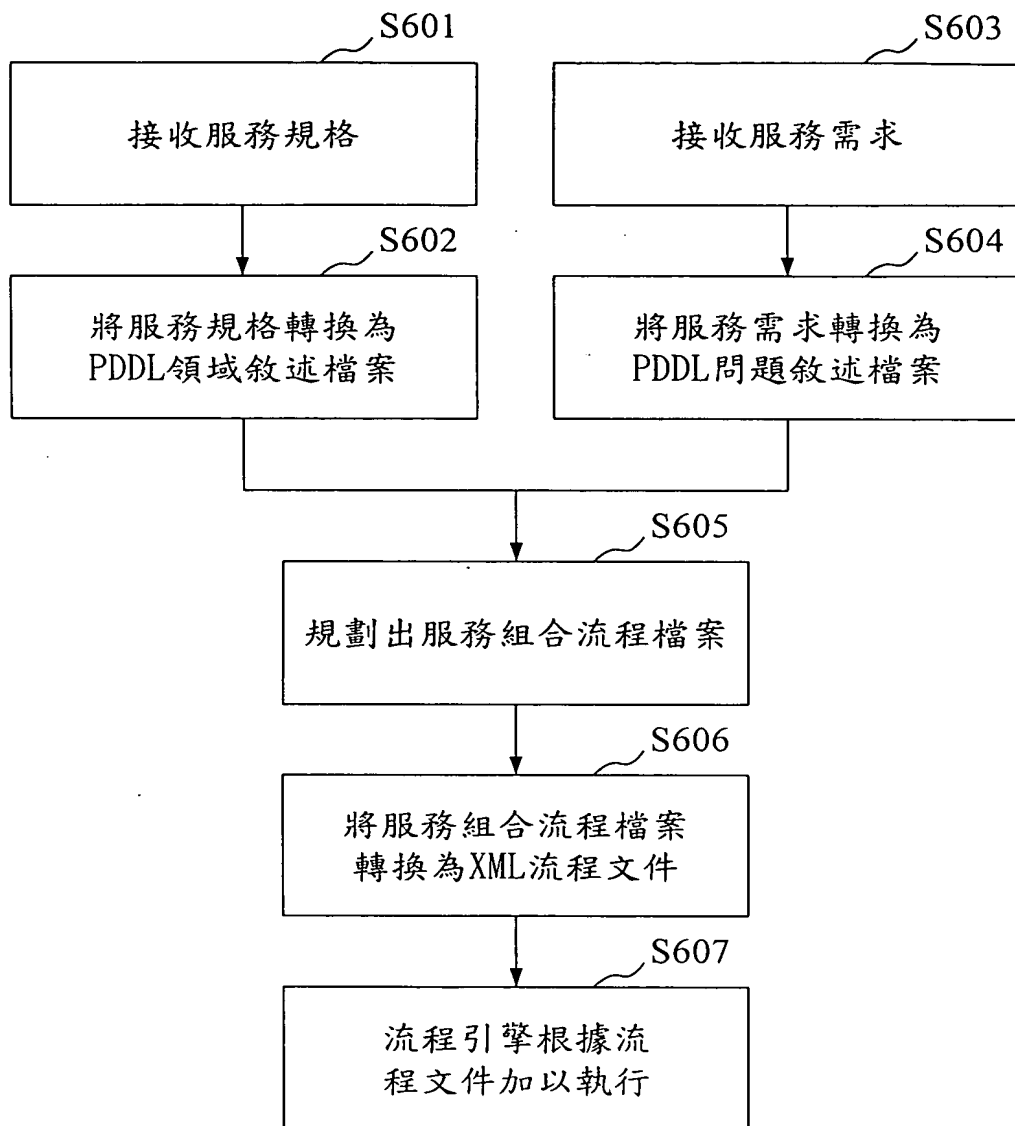


圖6

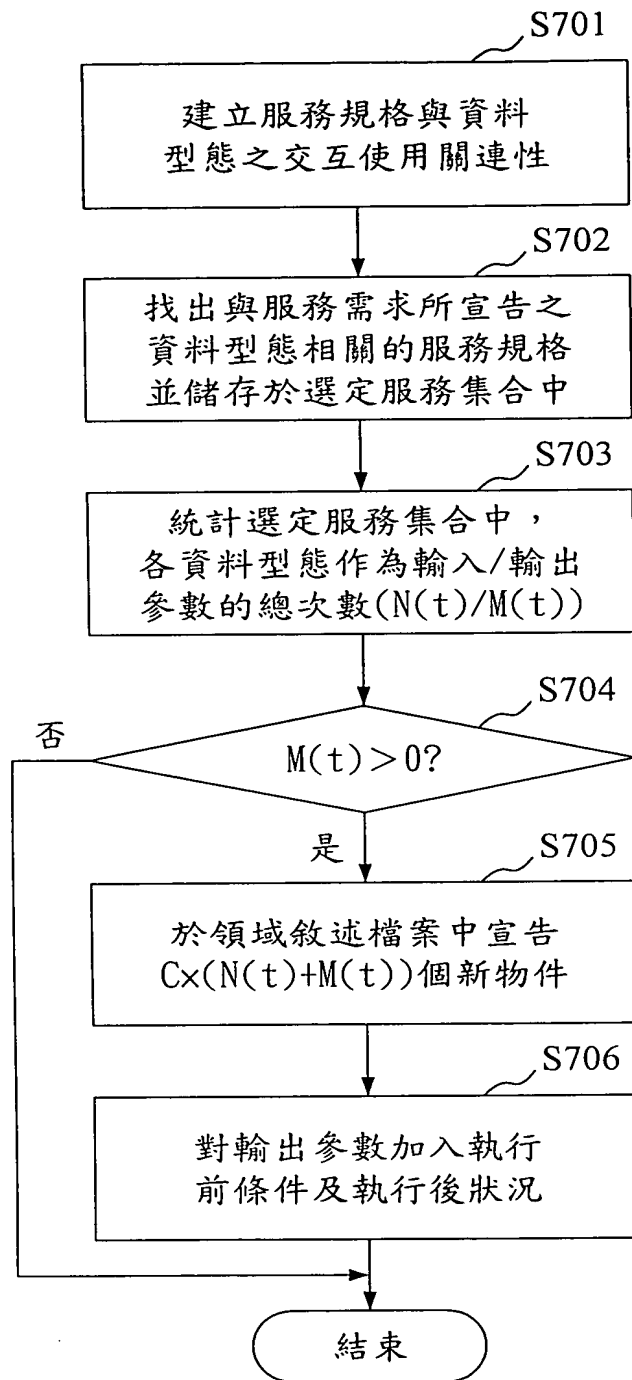


圖 7

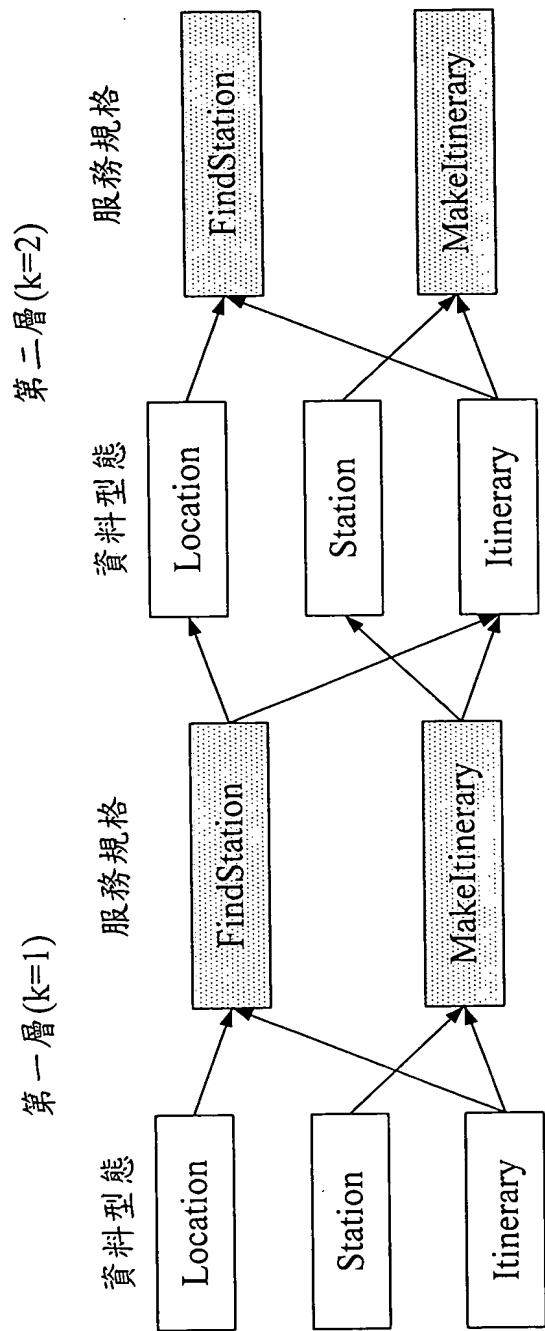
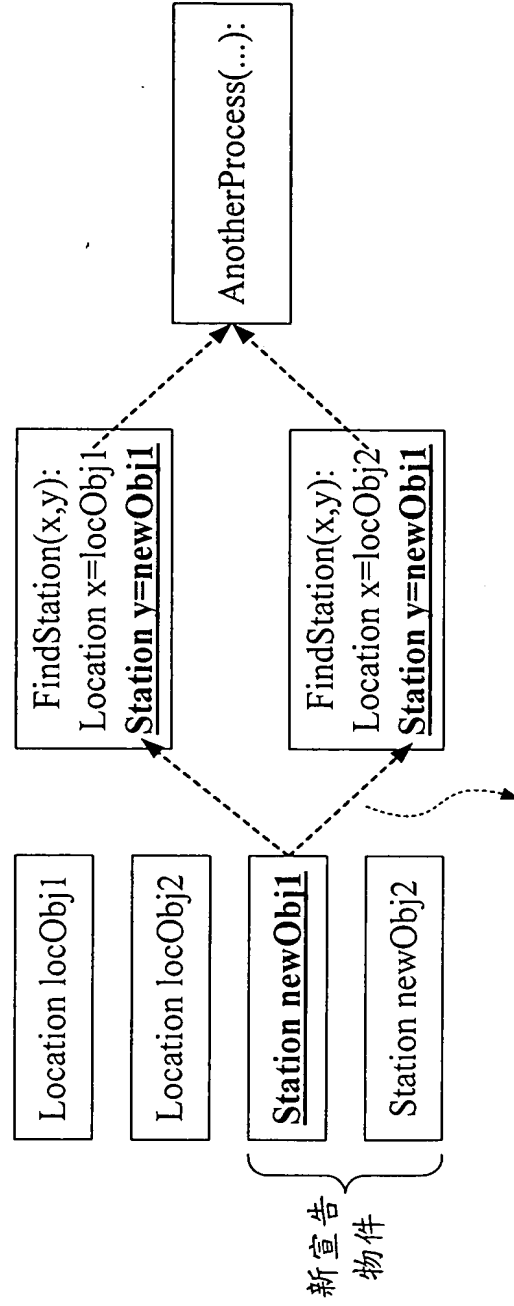


圖8



(重複利用新宣告物件)

圖9

PDDL問題敘述檔案程式碼

```
(define (problem iTour1)
  (:domain iTour)
  (:objects locObj1 locObj2 - Location
            itineraryObj - Itinerary)

  (:init (Start    locObj1)
         (End      locObj2))
  (:goal (and (Trip itineraryObj locObj1 locObj2))))
```

圖 10

PDDL領域敘述檔案程式碼

```
(define (domain iTour)
  (:requirements :strips :equality :typing)
  (:predicates
    (Start ?loc - Location)
    (End ?loc - Location)
    (At ?station - Station
      ?loc - Location)
    (Trip ?i - Itinerary
      ?loc1 ?loc2 - Location)
  )

  (:action AskStation
    :parameters (?loc - Location
      ?station - Station)
    :effect (and (At ?station ?loc))
  )

  (:action MakeItinerary
    :parameters (?loc1 ?loc2 - Location
      ?station1 ?station2 - Station
      ?i - Itinerary)
    :precondition (and (Start ?loc1)
      (End ?loc2)
      (At ?station1 ?loc1)
      (At ?station2 ?loc2))
    :effect (and (Trip ?i ?loc1 ?loc2))
  )
)
```

圖 11

PDDL領域敘述檔案程式碼

```

(define (domain iTour)
  (:requirements :strips :equality :typing)
  (:predicates
    (Start ?loc - Location)
    (End ?loc - Location)
    (At ?station - Station
      ?loc - Location)
    (Trip ?i - Itinerary
      ?loc1 ?loc2 - Location)
    (used ?o - Object)
  )

  (:constants newStationObj1 newStationObj2 newStationObj3 - Station
    newItineraryObj - Itinerary)

  (:action AskStation
    :parameters (?loc - Location
      ?station - Station)
    :precondition (and (not (used ?station)) )
    :effect (and (At ?station ?loc)
      (used ?station)
    )
  )

  (:action MakeItinerary
    :parameters (?loc1 ?loc2 - Location
      ?station1 ?station2 - Station
      ?i - Itinerary)
    :precondition (and (Start ?loc1)
      (End ?loc2)
      (At ?station1 ?loc1)
      (At ?station2 ?loc2)
      (not (used ?i))
    )
    :effect (
      and (Trip ?i ?loc1 ?loc2)
      (used ?i)
    )
  )
)

```

圖 12